**Lets stop and talk about storage and files**

Lets take a break, and quickly talk about something that not many developers understand ...

**Multiple files are stored on your computer when you visit a site**

I want us to quickly chat about what’s stored on your device (phone or PC) and browser every time you visit an app.

The starting point is to realize that multiple files travel between the browser and a web server when you request a web page by typing a URL in the address bar.

**DNS Files**

The first files that are stored on your computer are **DNS files**. Don’t worry, these are just files that refer to the domain name (in case you didn’t know, the domain name is always converted to an IP address). Interesting side note, when a browser looks up a domain name, multiple IP addresses could be returned as it has to hop around to find that particular page.

Ok ok Clyde. So these DNS files are stored on my browser, but for how long? It varies from browser to browser, but generally those records are stored up to 30min. Not only that, but your operating system may have a DNS cache of its own, separate from your browser. The time that data is stored largely varies depending on the website host.

Files pointing to a particular web page aren’t the only DNS files that are stored — your computer will also store DNS files for the items on that web page, such as photos and videos.

**Do you ‘cache’ my drift**

Once you make it to a web page, your **browser’s cache** will store a host of different files (things like HTML files, CSS style sheets, JavaScript code, and even images and videos). As you can imagine, that data adds up, fast.

Your browser uses cache because it wants to help you. When you revisit a web page later, the browser can check what kind of files are on the page that were previously downloaded, and only download the files that weren’t previously downloaded.

Isn’t this dangerous? Can your browser end up storing trillions of files that clog up your memory? No. Don’t stress. Browsers won’t just download an unlimited number of files. Generally, browsers cap out at a certain amount of data — and when that limit is reached, older files are deleted to make room for newer ones.

**old-school Cookies**

Cookies are also stored by websites (the website you visit will have server-side code that tell the browser to use cookies). When you visit a website, a cookie can be sent to your browser from that website’s server. The goal of a cookie is to act like an identifier — so that web page and others can serve up personalized information. What kind of information? Well, information like your name, age, email address and more.

There are 2 kinds of cookies — session cookies and persistent cookies. Session cookies are erased as soon as you exit the web browser. They’re stored in your computer’s memory, and do not collect information from your computer. Persistent cookies, however, is stored on your computer’s hard drive until it either expires or you delete it. These are the types of cookies that you’ve probably most heard about — they identify things like web browsing behaviour and user preferences, and as such can be used to advertise to you.

**Important:**cookies are outdated now. I just wanted to mention them as most of you have probably heard of them. Today, we have newer and more effective APIs for storing client-side data like the Web Storage API and IndexedDB API. Some modern browsers also support the new Cache API.

As a side note, the IndexedDB is an in-browser document database. This means that with IndexedDB you can create, read, update, and delete large sets of records in much the same way you are accustomed to doing with server-side databases. Databases are a huge topic of their own so I'll leave it here. I just wanted you to be aware of it.

**XMLHttpRequest object**

You know that DNS files are stored, cached information is stored, as well as cookie information. But what about when you request data via an XMLHttpRequest?

Good question.

The starting point is that if you’re using XMLHttpRequest to get data from a website, it has to be stored somewhere right? Otherwise, you would not be able to access it. Makes sense.

When you use the XMLHttpRequest to retrieve data, its stored on a temporary internal URL that points to an object stored inside your browser. In simple terms, **your browser stores the response information** automatically.

Don’t let this scare you.

At the crux its not much different from any variable we use to store information, like when we store our dog name in a variable like this: let myDogName = “woofey”. A variable is a “named storage” for data. If you’re not sure what a variable is, please check out my [JavaScript Complete Grandmaster course](https://www.udemy.com/course/new-javascript-grandmaster/?referralCode=758CCEBE390D73C4DC03).

Anyway, the point I’m trying to make is that your browser saves the variable name and associated data in memory. As to exactly where its stored, this is a very advanced topic so I won’t get into it here (I don’t want to get into the CPU register, volatile memory, or the stack, or a Context object … you can see there’s a lot to it).

Take a step back.

Just remember that if you can access data (as you can with a simple variable or the XMLHttpRequest object), then it has to be stored somewhere. Your browser handles all of this complexity in the background.

**Closing comments**

As you can see, there are a ton of files and data that are stored on your computer. Some are what most people have heard of, like cookies, cached data and DNS files.

**Bottom line: when you use the XMLHTTPRequest object to retrieve data, your browser also stores that data locally in its memory.**

Hope this has been interesting.

See you in the next lectures.